

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-291403

(43)Date of publication of application : 05.11.1993

(51)Int.Cl.

H01L 21/82

G06F 15/60

(21)Application number : 04-094074

(71)Applicant : HITACHI LTD

(22)Date of filing : 14.04.1992

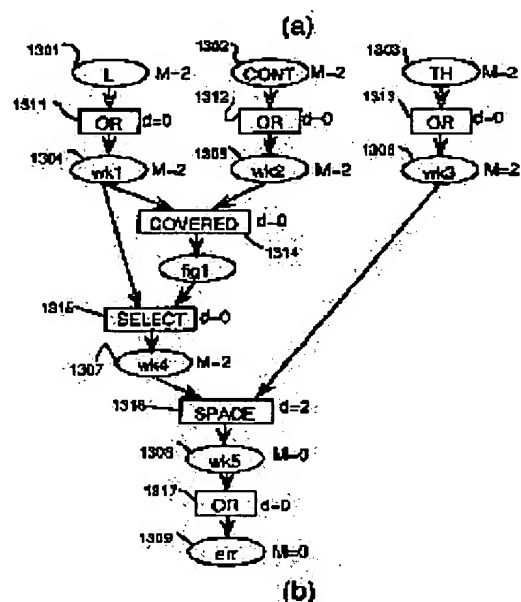
(72)Inventor : KO MASAOKI

(54) PATTERN DATA PROCESSING METHOD FOR SEMICONDUCTOR INTEGRATED CIRCUIT

(57)Abstract:

PURPOSE: To reduce the overhead of division control method by executing the data division and synthesis of result data for graphic arithmetic sequence to be processed only for the input data and output results of sequence.

CONSTITUTION: Boundary margin width during division is not expanded in the middle of sequence, and the margin width of L, CONT and TH is 2 in all cases. As a result of OR operators 1311 and 1312, layers wk 1 and wk 2 become input for COVERED to be executed in succession so that graphic number matching processing is performed. A COVERED operator 1314 requires no boundary margin resetting processing and form information matching processing is performed. A SELECT operator 1315 extracts only graphics of graphic number from a vector file wk 1. SPACE operator 1316 requires no form information matching processing and performs the boundary margin reducing processing for 2 only. Finally, OR operator 1317 performs graphic number arrangement processing and output result synthesis processing.



LEGAL STATUS

[Date of request for examination]

26.03.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3185342

[Date of registration]

11.05.2001

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11)特許番号

特許第3185342号
(P3185342)

(45)発行日 平成13年 7 月 9 日(2001. 7. 9)

(24)登録日 平成13年 5 月11日(2001. 5. 11)

(51)Int.Cl. ⁷	識別記号	F I
G 0 6 F 17/50	6 5 8	G 0 6 F 17/50
	6 2 6	6 5 8 M
H 0 1 L 21/82		6 5 8 L
		6 2 6 C
		C

請求項の数 5 (全 16 頁)

(21)出願番号	特願平4-94074
(22)出願日	平成 4 年 4 月14日(1992. 4. 14)
(65)公開番号	特開平5-291403
(43)公開日	平成 5 年11月 5 日(1993. 11. 5)
審査請求日	平成11年 3 月26日(1999. 3. 26)

(73)特許権者	000005108
	株式会社日立製作所
	東京都千代田区神田駿河台四丁目 6 番地
(72)発明者	高 正昭
	東京都国分寺市東恋ヶ窪 1 丁目280番地
	株式会社日立製作所中央研究所内
(74)代理人	100075096
	弁理士 作田 康夫

審査官 後藤 彰

(56)参考文献	特開 平 5 -249651 (J P, A)
	特開 平 4 -145585 (J P, A)
	特開 平 4 -43466 (J P, A)

(58)調査した分野(Int.Cl. ⁷ , D B名)
G06F 17/50
H01L 21/82

(54)【発明の名称】 図形パターンデータ処理方法

(57)【特許請求の範囲】

【請求項 1】半導体集積回路のレイアウトパターンに対し、電子計算機を用いて、図形演算子を組み合わせた図形演算シーケンスを施す方法であって、パターンデータを区分けして複数の部分領域を設定する境界決定処理を行い、前記境界決定処理により定められた境界に従って、各部分領域に属するように入力パターンデータを分割して複数の部分データからなるデータグループを生成するデータ分割処理を行い、処理すべき図形演算シーケンスを構成する図形演算子ごとに、入力となる前記データグループを構成する部分データのおおのについて図形演算を実行して部分結果データを求めることにより図形演算結果データグループを構成し、

境界処理が必要な場合は図形演算実行時に境界処理に必要な情報を抽出し、

該情報を用いて境界処理を実行して前記図形演算結果データグループを補正し、後続の図形演算子に前記図形演算結果データグループを受け渡す分割図形演算処理を行い、

出力すべきデータグループのおおののに関し、前記データグループを構成する各部分データを合成する出力結果合成処理を行なう、ことを特徴とする半導体集積回路のパターンデータ処理方法。

【請求項 2】各図形についてその近傍図形を考慮する必要がある図形演算子を含む図形演算シーケンスを処理する際に、

前記境界決定処理において、与えられた境界マージン幅だけ境界近傍で領域相互の重複を有するよう各部分領域

を拡張することを行い、
 前記分割図形演算処理における境界処理において、
 前記分割図形演算処理により不正確な境界近傍図形が生成される場合に、境界マージン幅を縮小して領域をはみ出る図形を除去し、
 後続の図形演算子に必要とされる値よりも境界マージン幅が小さい場合に、前記データグループを構成する各部分データに関し隣接する領域の部分データから不足する境界近傍図形を取得することにより境界マージン幅を拡張する境界マージン再設定処理を行う、
 ことを特徴とする請求項1記載の半導体集積回路のパターンデータ処理方法。

【請求項3】処理中の図形演算子が、面積値や位相情報のように部分領域内のみでは正確に取得できない形態情報に関する場合、

前記分割図形演算処理における境界処理において、前記データグループを構成する各部分データに関し境界近傍図形を取り分け、前記近傍図形に前記部分領域内での形態情報を付随させ、前記各部分データを用いて複数領域にまたがって存在する図形に関する形態情報を合成し、前記形態情報を用いて前記図形演算子の処理を完結する形態情報整合処理を行う、ことを特徴とする請求項1または2記載の半導体集積回路のパターンデータ処理方法。

【請求項4】前記各部分領域にそれぞれ別の処理装置を割り当て、前記各部分領域に対応する分割データの処理を並列に実行することを特徴とする請求項1乃至3のいずれかの請求項に記載の半導体集積回路のパターンデータ処理方法。

【請求項5】電子計算機を用いて、図形パターンデータを複数の部分領域に分割し、該電子計算機内の複数の処理装置のそれぞれに、前記各部分領域に分割された図形パターンデータを割り当てて、図形演算子を組み合わせた図形演算シーケンスを施す方法であって、
 前記電子計算機のマスタ処理装置が、入力された図形パターンデータを区分けする複数の部分領域を設定する境界決定処理を行い、
 前記境界決定処理により定められた境界に従って、各部分領域に属する図形パターンデータを分割して複数のデータグループを生成するデータ分割処理を行い、
 前記各データグループを前記複数の処理装置のそれぞれに割り当てて配分し、
 前記複数の処理装置のそれぞれが、独立して、
 前記図形演算シーケンスを構成する図形演算子ごとに、前記割り当てられたデータグループを構成する部分データのおおのについて図形演算を実行して部分結果データを求めることにより図形演算結果データグループを構成し、
 境界処理が必要な場合は前記図形演算実行時に境界情報を抽出し、前記境界情報を用いて前記マスタ処理装置が

境界処理を実行して、その結果に基づいて対応する処理装置が前記図形演算結果データグループを補正し、
 後続の図形演算子に前記図形演算結果データグループを受け渡す分割図形演算処理を行い、

前記各処理装置が分割図形演算処理を終了した前記図形演算結果データグループを前記マスタ処理装置に転送し、

前記マスタ処理装置が前記各図形演算結果データグループを構成する各部分データを合成する出力結果合成処理を行なう、ことを特徴とする図形パターンデータ処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は電子計算機を利用した図形パターンデータの処理方法に関わり、特に半導体集積回路のレイアウトパターンデータ等を自動的に処理する電子回路自動設計方法に適用して有効な技術である関する。

【0002】

【従来の技術】従来、半導体集積回路のパターンデータは逐次処理型の汎用計算機で一括して図形演算処理されていた。なお、図形演算とは、論理演算、サイジング演算、寸法検証演算、パターン面積検証演算、パターン周辺長検証演算、接続検証演算、位相検証演算、台形分割演算、およびその類似の演算子の総称と約束する。これらの演算子の仕様については例えば、プロシーディング・オブ・インターグラフィックス'83のテクニカル・セッションB3-4の11ページに記載されている。

【0003】しかし、近年の半導体集積回路の飛躍的な規模増大により処理時間が年々増加し問題となっている。これに対し近年、パターン平面を複数の部分領域に分割して各領域ごとにパターンを処理して結果を合併することにより所望の結果を合成することが行われている。このような方法は一般に、分割統治法と呼ばれている。

【0004】分割統治法は処理を高速化する上で以下の3つの効果がある。まず、処理の計算量のオーダが処理データ量に対し比例よりも大きい場合、分割処理の処理時間の総和は一括処理の場合よりも短くなることが期待できる。また、一度に処理すべきデータ量が小さくなるので、大容量であるが低速な記憶装置（例えば磁気ディスク）でなくても、高速であるが容量の小さい記憶装置（例えば半導体記憶装置）を用いてデータ処理を行なえるので、データアクセスを高速化できる。さらに、各分割データを別の処理装置に割り当てて並列に処理を実行でき、1つの分割領域の処理時間で全体の処理を行える。

【0005】分割統治法では分割境界に関する補正処理（以下、境界処理と呼ぶ）が重要となる。例えば、レイアウト規則検証処理のように、注目図形から設計規則バ

ラメタを適用する近傍領域を抽出する必要がある図形演算では、領域境界近傍において処理が正確に行われないため、これを補正する必要がある。また論理演算では、連結した1個の図形には同一の図形番号が付されなければならないが、複数部分領域にまたがる図形については、単に論理演算を各分割領域ごとに行なうだけでは、領域ごとに別の図形番号が付されてしまうため、これも補正が必要である。

【0006】このような境界処理に関し、例えば特開平2-125369では、通常の分割領域の他に境界近傍領域をまとめて一つの領域として別個に処理を行うことにより境界近傍のレイアウト規則検証を正確に行う方法、および、分割時に領域境界で切断された図形に関して接続情報を別途保存しておき、これと分割データの処理結果から全領域での図形の接続情報を合成する方法が述べられている。

【0007】

【発明が解決しようとする課題】分割統治法による従来のパターンデータ処理方法では、1個の図形演算子の高速化に焦点を当てた境界処理方法をとっており、複数の図形演算子からなる図形演算シーケンスを処理するという視点が欠落していた。

【0008】先に述べた従来技術では、境界近傍図形や接続情報という境界処理に必要な情報（以下、境界情報と呼ぶ）をデータ分割時に参照しているため、複数の図形演算子間でデータを受け渡すためには、受渡しデータに関する境界情報を生成し、これを参照する必要がある。また、図形演算子の種類によって分割データの保有形式が異なるため、異種の図形演算子間でデータの受渡しには分割データの保有形式の変換が必要である。従来技術によって上記の処理を行うためには、受渡しデータを一度マージした後再分割する必要がある。例えば、データに論理演算を施した後さらに寸法検証演算を施す場合、従来技術の第2の方法によりデータ分割、論理演算および結果合成を行った後で、第1の方法でデータ分割、寸法検証演算および結果合成を行わなければならない。

【0009】このように、従来の分割統治法では、図形演算シーケンスの処理は、1個の図形演算子に関する分割統治法を積み重ねることにより実現していた。しかし、データ分割および結果合成処理は、おのおのの入力および結果データの全量を逐次処理する必要がある、かなりの処理時間を必要とする。そして実際のアートワーク処理は複数の図形演算子により構成されるため、従来の方法では十分な高速化を実現することができなかった。

【0010】特に、複数の処理装置を用いて図形演算を並列に行う場合、分割数を増やすと先述の3つの効果により図形演算処理時間が減少するが、処理時間のかなりの部分をデータ分割と結果合成で費やされることになる。そのため、処理装置の台数を増やしても、台数に見

合った高速化が達成できないという問題が存在する。また、従来の分割統治法では、パターンの面積検証、パターン周辺長検証あるいは位相検証演算などの図形演算子に対応できなかった。

【0011】本発明は上記の問題点を克服するものであり、高速であり、かつ任意の図形演算シーケンスを処理する汎用性を有する、分割統治法に基づいたパターンデータ処理方法を提供することを目的とする。

【0012】

【課題を解決するための手段】上記の目的を達成するため、分割統治法に基づく本発明のパターンデータ処理では、入力となるデータグループを構成するそれぞれの部分データについて図形演算を実行して部分結果データを求めて図形演算結果データグループを構成し、さらに、境界処理が必要であれば図形演算実行時に境界処理に必要な情報を抽出し、これを用いて境界処理を実行して上記図形演算結果データグループの内容を補正することにより、複数の図形演算子間で分割データを直接受け渡すことを可能とする。特に、境界情報は図形演算実行時に参照され、分割データの保有形式は演算子によらず同一とすることにより、図形演算子間での分割データの受渡しを可能としている。

【0013】各種図形演算子に関する境界処理方法は具体的には次のとおりである。寸法検証演算子などの様に、各図形についてその近傍図形も考慮する必要がある図形演算子を含む図形演算シーケンスの場合、以下の処理を行なう。まず、境界決定処理において、境界近傍で領域相互が重複するように所定の境界マージン幅だけ各部分領域を拡張する。さらに、分割図形演算処理において、演算処理によって境界近傍で不正確な図形が生成される場合に、その演算子に関する分割結果データの出力時、あるいは、その演算結果を入力とする演算子に関する分割データの入力時に分割データの境界マージン幅を縮小し、不正確な境界近傍図形を除去する（境界マージン縮小処理）。また、図形演算結果データグループの境界マージン幅が後続の図形演算子に必要な値よりも小さい場合、上記データグループを構成する各部分データに対応する領域に隣接する領域の部分データから不足する境界近傍図形を付加することにより境界マージン幅を拡張する（境界マージン拡張処理）。

【0014】また、処理中の図形演算子が、部分領域内に関する情報だけでは構成できない面積値や位相情報のような形態情報に関係する場合、以下の境界処理を行う。まず、図形演算の実行時に、図形演算結果データに関する境界近傍図形を取り分け、これを部分領域内の形態情報として格納する。次に、境界処理の際、この近傍図形とその形態情報を用いて複数領域にまたがって存在する図形に関する形態情報を合成し（形態情報整合処理）、この形態情報を用いて複数領域にまたがって存在する図形に対するレイアウト規則の判定などを行なっ

て、図形演算子の処理を終了する。

【0015】

【作用】本発明の方法によれば、複数の図形演算子からなる図形演算シーケンスを分割統治法により処理する際に、各図形演算子ごとに境界処理が必要な場合に、図形演算の実行時に境界近傍図形を取り分け、境界近傍図形に関する形態情報を求め、これらを用いて境界処理時に図形演算結果データグループの内容を補正する。これにより、任意の図形演算子に関して、その分割結果データグループの内容は領域全体において正確となり、図形演算子間で分割データを直接受け渡すことができる。従って、データ分割は演算シーケンスへの入力となるデータについてのみ行えば良く、データマージは演算シーケンスからの出力となるデータについてのみ行えば良いので、分割統治法のオーバーヘッドを最小限にでき、高速な処理実行が可能である。

【0016】特に、各分割領域に対してそれぞれ別の処理装置を割り当てて演算処理を並列に実行する分割統治法の場合、図形演算処理がほぼ部分領域1個分の処理時間で実行可能となり、処理全体に対する比重が大きいデータ分割およびデータマージの処理時間が削減されるので、本発明による図形演算はさらに高速化される。

【0017】

【実施例】実施例の説明の前に、準備として図形演算について説明する。図14(a)に示すように、図形演算で取り扱う図形データは、図形1401は外縁辺をベクトルで表現し図形番号とベクトルの向きを付した形式(1402、1403)で表現される。ここでベクトルの向きは、図形内部を進行方向に向かって右手に見るように定義される。ベクトル形式の図形表現では通常、図中の垂直方向のベクトルは省略される。上記の図形表現により任意の多角形データを表現できる。上記の図形データは具体的には図14(b)に示すように、始点終点形式1404(レコード形式：順に、図形番号、ベクトルの始点のx座標、y座標、終点のx座標、y座標)あるいは、左右端点形式1405(レコード形式：順に、図形番号、左側端点のx座標、y座標、右側端点のx座標、y座標、ベクトルの向き(左右)を示すフラグ)により記述され、いずれの形式も使用できる。

【0018】図形演算子には、先に述べたように論理演算、サイジング演算、寸法検証演算、パターン面積検証演算、パターン周辺長検証演算、接続検証演算、位相検証演算、及び台形分割演算があるが、このうち、以下の説明で使用する演算子のみ、その処理内容を詳細に説明する。以下では、AL層などの物理的意味を有するデータに限らず、図形演算結果の図形データも含めたデータの集合を層と呼ぶ。

【0019】論理演算子は、1つ又は複数の層に対して集合演算を行い、その結果を出力する。論理演算子では、出力図形の輪郭取り(図形の重なるの削除)が行わ

れ、連結した図形には同一の図形番号が付される。図15(a)、(b)には入力図形の集合和を求めるOR演算子の入力層と出力層の一例が示されており、入力層1501では図形番号#13と#16の図形は連結しているので、出力層1502では1個の図形として#5という図形番号が新たに与えられている。

【0020】次に、位相検証演算子について説明する。位相検証演算子は、輪郭取りされた2つの層A、Bを入力し、それぞれA、Bに属し、かつ互いに所望の位相関係(交差、一致、包含、被包含など)にある図形を1個ずつ組み合わせた図形対をもれなく求めて、その図形番号対を出力する演算子である。図15(c)、(d)では被包含の位相関係を抽出する位相検証演算子(COVERED演算子)の入力層と出力層の一例が示されており、この演算子は図形B(1503)に完全に含まれる図形Aをすべて求めてその図形番号の対1504を出力する演算である。位相検証演算子では図形番号に基づいた処理がなされるので、入力層において連結した図形にはユニークな図形番号が付されていないなければならない。

【0021】最後に寸法検証演算子について説明する。寸法検証演算子は、輪郭取りされた1個あるいは2個の層を入力して図形の幅や図形間の間隔を調べ、幅が細い図形や微小間隔をもれなく求める演算子である。図15(e)、(f)では微小間隔を求めるSPACE演算子の入力層と出力層の例が示されている。図では、2つの輪郭取りされた層A、Bを入力してAとBの各図形の間隔を検証し、与えられたパラメタdよりも間隔が狭い上記違反地点近傍の図形を抽出して出力する演算である。図15(e)、(f)の例では入力層1506には1508に示す間隔の狭い部分が存在するので、この狭い間隔1508が抽出結果となり、SPACE演算子の出力層は1507のようになる。

【0022】以上の準備のもとに、以下、実施例を説明する。図1は本発明の一実施例の処理フローを示した図である。図1に示される実施例は大きく4つの処理、すなわち、境界決定処理101、データ分割処理102、分割図形演算処理103、及び出力結果合成処理104から構成される。それぞれの処理の詳細について以下に説明する。

【0023】図2は境界決定処理101の詳細処理フローを示した図である。まず、図形のX座標、Y座標それぞれの最小値、最大値を求めることにより、データの存在範囲を求める(201)。次に、この存在領域全体を互いに重ならない複数の部分領域に分割する(202)。図3(a)に示すように、分割は、一次元分割(301)、二次元分割(302)、またはデータの疎密に応じた分割(303)が可能である。分割303を行うには例えば、プロシーディング・オブ・インターナショナル・コンファレンス・オン・コンピュータ・エイデッド・デザイン・1988の278ページに記載の方

法を使用できる。以下の実施例では、簡単のために次元分割の場合を説明するが、本発明は二次元分割の場合にも適用可能である。最後に、処理すべき演算シーケンスが寸法検証演算子などのように各図形の近傍にある図形を考慮する必要のある図形演算子（以下、近傍考慮演算子と呼ぶ）を含む場合、領域間の境界マージンが必要となるので、図3（b）に示すように各分割領域を304から305へ拡張する（203）。以下では、境界304を真境界、境界305を仮境界、拡張幅306を境界マージン幅と呼ぶ。仮境界305、及び境界マージン幅306は後述の境界マージン再設定処理によって変更される場合もあるが、真境界304は処理中は不変である。境界マージンを設定しない場合には仮境界は真境界と一致し、境界マージン幅は0である。境界マージン幅の決定方法については処理全体を説明した後で説明する。

【0024】データ分割処理102では、境界決定処理101で決定した仮境界に基づいて入力図形データを分割して、複数の部分データからなるデータグループを作成する。ここで、複数の領域にまたがる図形は仮境界の外側の部分は切り捨てられる。境界マージンが設定されている場合、境界マージン内の図形は複数の部分データに含まれる。なお、並列処理型の分割統治法の場合、複数の部分データは対応するそれぞれの処理装置に分配される。

【0025】続いて、処理すべき図形演算シーケンスを構成する図形演算子ごとに、分割図形演算処理103を行う。図1の中に示されている分割図形演算処理の概略フローを用いて処理の概略を説明する。まず、入力となるデータグループを構成する各部分データに対して非分割の場合と同様の図形演算105を実行し、演算結果である部分データを求める。さらに、演算子が近傍を考慮すべき演算子である、あるいは面積や位相情報のように部分領域内のデータだけでは正確に構成できない形態情報に関係する演算子（以下、形態考慮演算子と呼ぶ）である場合、境界処理106が必要となる。境界処理106が必要な場合には、その処理に必要な境界情報を図形演算時に合わせて求め出力する（境界情報取得処理）が必要になる。境界処理106は境界マージン再設定処理と形態情報整合処理に分類される（双方の処理を行う場合もある）。以下、上記処理の詳細を説明する。

【0026】まず、境界マージン再設定処理は、近傍考慮演算子において境界マージン幅を縮小して仮境界近傍にある演算結果の図形のうち不正確な部分を切り捨てる境界マージン縮小処理、あるいは、近傍考慮演算子が後続する場合に隣接する領域から仮境界近傍の図形データを獲得して境界マージン幅を拡張する境界マージン拡張処理である。ところで、負の境界マージン幅は禁止される（真境界近傍の図形が失われてしまうため）ので、近傍考慮演算子の入力データグループの境界マージン幅は

その演算子が考慮すべき近傍幅以上でなければならない。そのため、処理すべき演算シーケンスを構文解析することにより、最初のデータ分割時の境界マージン幅および、各演算子ごとの境界マージンの再設定仕様（再設定しない演算子が存在しても良い）を決定することが必要となる。境界マージンの設定方法は後で詳細に説明する。

【0027】まず、境界マージン縮小処理は、図4に示すように、単に新しい領域からはみ出る図形を切断し（401）、残った部分を出力（402）すればよいので、図形演算の部分結果データ出力時、あるいは後続の図形演算子のデータ入力時のいずれの時点で行なっても良い。図5は境界マージン縮小処理を具体的に説明した図であり、仮境界を501から502に縮小する場合の入力（a）と出力（b）の例が示されている（503は真境界）。

【0028】次に、図6、7を用いて境界マージン拡張処理の詳細処理フローを説明する。図6（a）は境界マージン拡張処理のための境界情報取得処理の詳細フローであり、図6（b）はその出力データである。まず、図形演算処理から出力される図形ごとに真境界からはみ出した部分を切除する（601）。次に、残った部分について、真境界から新たに設定すべき境界マージン幅だけの内り領域を設定し、この内り領域にかかるベクトルについては該当する境界に関する境界データファイル（左右の境界について別々に存在）に出力し（602）、内り領域にかからないベクトルについては非境界データファイル604に出力する（603）。ここで、両側の内り領域に入るベクトルは、2つの真境界の中央で切断して、各々を左右の境界データファイルに出力する（中央切断処理）。ところで、図6（b）に示される出力データは1個の分割領域に関する出力であり、これらは分割領域の数だけ独立に出力される。それぞれの出力データを区別するため、データファイル名の後に対応する領域の番号を付加して、境界データファイル#1などのように識別する。

【0029】次に、図6（b）の境界情報取得処理の出力データを受け取って実行される、境界マージン拡張処理における境界処理の詳細フローを図7に示す。まず、境界ごとに、境界の左右に該当する境界データを入力して、ステップ701から703より構成される境界OR処理を実行することにより、真境界で切断されたベクトルを接合して新境界データファイルに書き出す処理、および複数領域にまたがって存在する図形に関する図形番号マージ情報を求める処理を行う。ここで、真境界に接触しないベクトルはそのまま新境界データファイルに出力される。次に、ステップ704では、図形番号マージ情報を入力して、複数領域にまたがる図形にユニークな図形番号を付与するための図形番号変換表を作成する。ステップ701から704の処理はOR演算と同様の算

法で実行される。次に、ステップ705と706では、各新境界データファイルについて、新しい境界マージンに従ってベクトルを分割し分配する。その結果、各非境界データファイルには拡張された境界マージン領域内のベクトルが格納される。最後に、ステップ708では、先に求めた図形番号変換表を用いて各結果データの図形番号を付け直す。

【0030】図8を用いて、図7に示した境界マージン拡張処理の具体例を説明する。図8(a)は、2分割の場合に境界マージン幅0の分割データを入力して論理演算子を実行した結果のデータグループであり、領域#1(811)及び領域#2(812)に対応して2つの部分結果データが出力されている。図8(a)では、真境界816を境にして領域#1と領域#2は重ならず分割されている。また、数字815は各図形に与えられた図形番号である。他の領域の図形と図形番号を区別するため、出力図形の図形番号に領域の通し番号に十分大きい値(本実施例では10000)を乗じた値を加える。その結果、領域#1の図形には10001から10003の図形番号が付され、領域#2の図形には20001から20003の図形番号が付される。次に、図8

(a)の処理結果のうち境界線813、814により境界が示されている内り領域に対して境界情報取得処理を施した結果が図8(b)であり、非境界図形データ821、822と境界データ823、824に分けて出力される。次に、境界データ823、824を入力して境界マージン拡張処理における境界処理を行った結果が図8(c)である。ここで、例えば図形10003と20001とが連結していることを認識し、これらの図形を接合する処理を行って、新境界データ831、及び図形番号変換表832を作成する。図形番号変換表832の左右の欄はそれぞれ変換前後の図形番号を示す。例えば変換前の図形番号10003と20001の図形は連結しているので、変換後の図形番号はともに3となっている。表832には内り領域に含まれない(即ち、非境界データに含まれる)図形番号も格納されているが、これは変換後に欠番のない一連の番号(図8(c)では1から4まで)を付与するためであり、その必要のない場合には非境界データである変換前の図形番号が10001と20003の欄は省略できる。最後に、新境界データ831を境界813、814に従って非境界データ821、822に分配して(境界814より左の部分を非境界データ821へ分配し、境界813より右の部分を非境界データ822へ分配する)結果データを作成し、さらに図形番号変換表832に従って結果データの図形番号を変換して、図8(d)に示す所望の結果データ841、842を得る。

【0031】一方、形態情報整合処理は、仮境界近傍に存在する図形に関する不正確な形態情報を補正する。形態情報としては図形番号、面積値、周辺長値、位相情報

などがあり、寸法値(幅、間隔)は境界マージンが設定されているのでその値は正しく求められている。形態情報整合処理は図形演算子の種類によってその処理方法が異なり、それに対応して境界情報取得処理の処理方法も異なる。これらの処理のうち、論理演算子に関する形態情報整合処理についてはすでに境界マージン拡張処理のところで説明した。その他の演算子については、そのすべてについて形態情報整合処理の詳細フローを述べることはせず、代表的な位相検証演算子についてのみ説明する。ただし他の演算子についても、形態情報整合処理の処理方法は位相検証演算子の場合とほとんど同一である。

【0032】以下では図9、10、11を用いて、位相検証演算子のための形態情報整合処理の詳細処理フローを説明する。図9(a)は位相検証演算子の形態情報整合処理のための境界情報取得処理の詳細フローを示した図であり、図9(b)はその出力データとそのレコード形式を示した図である。図形演算処理で検出され、互いに交差する図形対の各々について、両図形が領域の境界マージン領域と接触あるいは交差しない場合には、両図形が所望の位相関係にある場合のみ、図形番号対をレコード形式924で非境界データファイル921に出力し(911)、それ以外の場合は、境界図形番号ファイル922に境界マージン領域と接触あるいは交差する図形についてその図形番号と層番号をレコード形式925で出力し、境界データファイル923に図形番号対とその位相コードをレコード形式926で出力する(912)。ここで境界図形番号ファイルと境界データファイルへの出力では境界(左右)の区別をしない。図9

(b)に示される出力データは1個の分割領域に関する出力であり、これらは分割領域の数だけ独立に出力される。上記の位相コードは、図9(c)に示すような層Aと層Bの図形の図形対に与えられる3bitのコード931であり、各ビットは上位bitから順に、両図形の交差部分が存在するか否か、図形Aのみの部分が存在するか否か、図形Bのみの部分が存在するか否かを表す。上記3bitの組合せによって、表932に示す8種類の2図形間の位相関係が定義される。

【0033】次に、図9(b)の境界情報取得処理の出力を受けて行う形態情報整合処理における境界処理の基本的な考え方を述べる。ある図形対について領域#iの位相コードはT1であり、領域#(i+1)の位相コードはT2である時、領域#iと#(i+1)の和領域における位相コードはT1とT2の各bitごとの論理和となる。従って全領域について位相コードのbitごとの論理和をとれば、全体の位相コードが合成できる。この処理を実行する詳細フローが図10に示され、上記処理で使用されるテーブルの仕様が図11に示されている。まず、境界データ#0、境界図形番号データ#0を各々テーブル#1、#3にセットする(1001)。な

お各テーブルは、検索に便利のようにデータをセットする段階で層番号ないし図形番号をキーとしてソートしておく。テーブル#1(1101)、#3(1103)は各々現在注目している境界の左側の領域に存在する図形対に関する位相コード及び出現領域番号を保持するテーブルである。次に、各境界ごとにステップ1002から1006を行う。境界データ#i及び境界図形番号データ#iを読み込んで、各々テーブル#2、#4にセットする(1002)。テーブル#2(1102)、#4(1104)は各々現在注目している境界の右側の領域における図形対の位相コードおよび図形番号を保持する。テーブル#1とテーブル#2の両方に格納されている図形対について、その位相コードのbitごとの論理和をとってテーブル#1に格納し、テーブル#2のレコードは削除される(1003)。テーブル#1にのみ存在する図形対については、領域#(i+1)での位相コードをテーブル#4から回復し(最上位bitは0とし、他のbitは、該当する図形がテーブル#4に存在すれば1とし、なければ0とする)、位相コードの各bitごとの論理和をとってテーブル#1に格納する。領域#(i+1)での位相コードが000の場合、その出現領域#から#iまでの新境界データファイルにテーブル#1のレコードをレコード形式926で出力した後、テーブルから削除する(1004)。さらにテーブル#2にのみ存在する図形対については、領域#iにおける位相コードをテーブル#3から回復し、位相コードのbitごとの論理和をとった後でテーブル#2のレコードをテーブル#1に登録する(1005)。最後に、テーブル#4の内容をテーブル#3に反映させる(1006)。上記ステップ1002から1006を各境界ごとに繰り返すことにより、複数領域にまたがる図形を含む図形対に関する位相情報の合成が行われ、その結果が領域ごとに対応する新境界データファイルに格納される。最後に、それぞれの新境界データファイルの中から所望の位相関係を満たすもののみを取り出して対応する非境界データファイルにマージすることにより、結果となる分割データを得る(1007)。

【0034】図12に示すCOVERED演算子の場合を用いて、以上説明した位相検証演算子の形態情報整合処理を具体的に説明する。図12(a)は、位相検証演算子の入力となるデータグループを示す。境界マージ幅は0であり、二つの部分領域#1(1211)、#2(1212)が重ならず分割されている。図中で数字1213は図形の所属する層と図形番号を示す。例えば、A-#3は層Aに所属し、図形番号が#3であることを示す。これにCOVERED演算子と境界情報取得処理を施した結果を図12(b)に示す。真境界に接触しない図形については通常のCOVERED演算を行って非境界データ1221、1222が出力される。真境界に接触する図形を含み互いに交差する図形対はすべて

出力されて境界データ1223、1224となり、真境界に接触する図形については境界図形番号データ1225、1226が出力される。全ての図形対について位相コードを出力するとそのデータ量は膨大となるので、交差しない図形対(つまり位相コードの最上位bitが0)については位相コードを出力しない。例えば部分領域#1において、境界に接触する図形B-#12は図形A-#4、A-#5と交差するが図形A-#6とは交差しないので、境界データ1224では、A-#4、A-#5との図形対に関する位相コードのみが出力される。次に、データ1223から1226を入力して形態情報整合処理を行った結果を図12(c)(d)に示す。データ1231、1232は新境界データであり、COVERED演算子に対応する位相コードは101であることを考慮すると、結果データ1241、1242が得られる。

【0035】位相演算子と同様の方法を用いて、パターン面積検証、パターン周辺長検証、接続検証などの形態考慮演算子を分割統治法により実行することができる。例えばパターン面積検証演算子での形態情報整合処理は、位相コードの代わりに分割領域(境界マージンを除く)内での図形の面積値を使用し、これを各領域について積算すれば良い。

【0036】全ての図形演算子の処理が終了すると、最後の出力結果合成処理104では、出力すべきデータグループのすべてについて形態情報の整合を取ったり、境界で切断された図形を接合するなどの境界接合処理を行い、部分データをまとめて1個のデータにする。この方法は上述の形態情報整合処理、境界マージン再設定処理と同様の方法で実現できる。

【0037】次に、図13(a)に示す図形演算シーケンスを例にとり、境界マージンの設定方法と各演算子の処理方法の一例を具体的に説明する。行1321から1327は図形演算式であり、左辺および右辺の図形演算子のオペランドに現れるLやwk1は層の名称を表わす。例えば、図形演算式1321は層LにOR演算子を実行し、その結果層をwk1と名付けることを意味している。図13(a)の図形演算シーケンスにおけるデータの流れを図13(b)に示す。図13(b)に示した楕円1301から1309は図形データを表わし、長方形1311から1317は図形演算子を表わし、矢印はデータの流れを表わす。

【0038】図13の図形演算シーケンスは、L層(1301)、CONT層(1302)、及びTH層(1303)の3層を入力して7つの図形演算子(1311から1317)を順に実行することにより、設計規則違反データerr1309を求めて出力する。図形データ(1304から1308)は中途結果である。図13の例では出力結果は1個のみであり、かつ、それは演算シーケンスにおける最後の演算子の結果でもある

が、一般には、出力結果が複数個存在する場合もあり、得られた出力結果を入力としてさらに演算を続けることもできる。

【0039】まず、図13の図形演算シーケンスを例として、分割時と各演算子における境界マージン幅の設定方法を説明する。ただし、以下の説明では境界マージン拡大処理は行わないとする。図13(a)のシーケンスにおいて、近傍考慮演算子はSPACE演算子1326のみである。先述のように近傍考慮演算子では、出力データにおいて指定した寸法パラメタ(1326のSPACE演算子では2)の幅だけ境界マージン幅を縮小しなければならない。一方、論理演算子や位相検証演算子では演算処理において近傍考慮をしないので、境界マージン再設定は不要である。データ分割時および各図形演算子の実行時に境界マージン幅を定める一般的な方法は、図13(b)のようなデータの流れに対して、データの流れを出力結果から入力データに向かって逆に辿りながら近傍考慮演算子の寸法パラメタを積算する。図13(b)の場合では、出力結果であるerr1309の境界マージン幅を0とし、データの流れの下流から順に必要な境界マージン幅を示すと、wk5(1308)は0、分岐を左に入ってwk4(1307)は2、さらに分岐を左に入ってwk1(1304)は2、L(1301)は2、COVEREDの分岐に戻ってwk2(1305)は2、CONT(1302)は2、SPACEの分岐に戻ってwk3(1306)は2、TH(1303)は2となる。なお、シーケンスの途中で境界マージン拡張処理を行う場合、その拡張幅は出力結果からその演算子までの寸法パラメタの積算値とし、そこで積算値を0としてさらにデータの流れをたどっていけば良い。

【0040】次に、本発明の分割統治法によって図13のシーケンスを処理する方法を具体的に説明する。境界決定処理とデータ分割処理は既に述べた通りである。分割時の境界マージン幅はシーケンスの途中で拡張せず、L、CONT、及びTHの境界マージン幅はいずれも2とする。OR演算子1311、1312の結果層wk1、wk2は続いて実行されるCOVERED演算子の入力となるため、図形番号整合処理が必要であり、すでに説明した図6から8の方法で処理される。OR演算子1313は形態情報整合及び境界マージン再設定処理ともに不要である。COVERED演算子1314は境界マージン再設定処理は不要であるが形態情報整合処理は必要であり、すでに説明した図9から12の方法で処理される。SELECT演算子1315は、図形番号ファイルfig1で与えられた図形番号の図形のみをベクトルファイルwk1から抜き出す演算であり、境界マージン再設定と携帯情報整合はいずれも不要である。続くSPACE演算子1316では形態情報整合処理は不要であるが、2だけの境界マージン縮小処理が必要であ

る。最後のOR演算子1317では図形番号整合処理と出力結果合成処理が必要であるが、その方法は既に述べた。

【0041】最後に、本発明の分割統治法によるパターンデータ処理方法を、複数の処理装置を用いて並列に実行する一例を説明する。図16は分割統治法による図形演算処理を並列に行う装置の構成を示す。処理装置はグローバル記憶装置1603が結合された1台のマスタ処理装置1601とそれぞれにローカル記憶装置1604が結合された複数のスレーブ処理装置1602とから構成される。それぞれの記憶装置は、メモリなどの揮発性の記憶装置と磁気ディスクなどの不揮発性の記憶装置の一方あるいは両方から構成される。各処理装置は通信路1605を介して結合され、相互に通信が可能である。図16では、通信路1605はバスのように一本の線という形状をとっているが、一般には格子結合、あるいはハイバキューブ結合など任意の位相構造をとることができる。図16に示した装置で、図1に示した分割統治法に基づく本発明の図形演算処理を並列に実行する方法を以下説明する。処理開始時には、入力データはグローバル記憶装置1603に格納されている。マスタ処理装置1601は入力データを読み込んで境界決定処理101を行い、各分割領域とスレーブ処理装置との対応を決定する。続いてデータ分割処理102を行い、分割データは通信路1605を介してスレーブ処理装置1602に送信され、ローカル記憶装置1604に格納される。次に、各スレーブ処理装置はローカル記憶装置に格納された分割データを読み込んで、それぞれが並行して分割図形演算処理を行う。境界処理106が不要の場合には各スレーブ処理装置はそれぞれ独立に演算子の処理を実行する。一方、境界処理が必要な場合には、各スレーブ処理装置は境界情報取得処理105までを行い、得られた境界情報をマスタ処理装置に転送する。マスタ処理装置は受信した境界情報を用いて境界処理を行い、境界処理の結果を各スレーブ処理装置に返送する。最後に、各スレーブ処理装置は返送されてきた境界処理の結果を用いて分割データの補正を行う。さらに具体的に説明すると、論理演算子に対する境界処理では、各スレーブ処理装置は境界情報取得処理までを行ってその結果データのうち境界データ605のみをマスタ処理装置に送信し、マスタ処理装置は図7に示した境界処理のうちステップ701から704までを行って図形番号変換表および新境界データを該当するスレーブ処理装置に返送し、最後に各スレーブ処理装置はステップ705から708をそれぞれ並行して実行する。境界処理の有無にかかわらず、分割図形に対する演算処理の結果である分割データは、それぞれ対応するローカル記憶装置に格納される。全ての図形演算子の処理が終了したら、各スレーブ処理装置は出力すべき分割データをマスタ処理装置に送信し、マスタ処理装置はこれを受け取って出力結果合

成処理104を行い、処理結果をグローバル記憶装置に格納する。

【0042】

【発明の効果】以上説明したように、本発明の半導体集積回路のパターンデータ処理方法では、分割統治法による図形演算処理の際、処理すべき図形演算シーケンスに対するデータ分割及び結果データ合成はそれぞれシーケンスの入力データ及び出力結果についてのみ実行すれば良く、分割統治法のオーバーヘッドを低減でき、高速な処理を実現できる。特に、複数処理装置を用いて並列図形演算処理を行う場合、上記の高速化の効果は顕著である。

【図面の簡単な説明】

【図1】本発明の一実施例を示す図である。

【図2】境界決定処理の詳細フローを示した図である。

【図3】境界決定処理の処理内容を説明するための図である。

【図4】境界マージン縮小処理の詳細フローを示した図である。

【図5】境界マージン縮小処理の処理内容を説明するための図である。

【図6】境界マージン拡張処理のための境界情報取得処

理の詳細フローおよびその出力データを示した図である。

【図7】境界マージン拡張処理のための境界処理の詳細フローを示した図である。

【図8】境界マージン拡張処理の処理を具体的に説明した図である。

【図9】位相検証演算のための境界情報取得処理の詳細フローおよびその出力データを示した図である。

【図10】位相検証演算のための境界処理の詳細フローを示した図である。

【図11】位相検証演算のための境界処理に使用されるテーブルの形式を示した図である。

【図12】位相検証演算のための境界情報取得処理および境界処理を具体的に説明した図である。

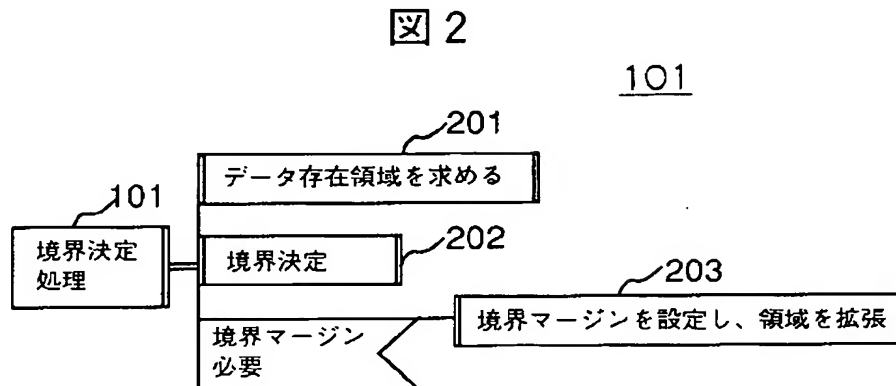
【図13】図形演算シーケンスの例を示した図である。

【図14】図形演算における図形の表現形式を説明するための図である。

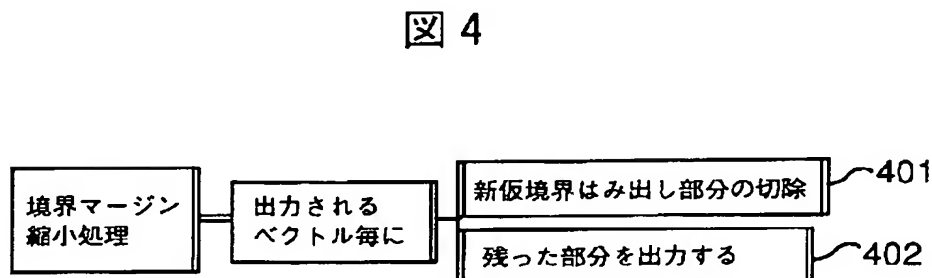
【図15】図形演算子の処理内容を説明するための図である。

【図16】複数処理装置を用いた、本発明を実施するための装置を示した図である。

【図2】

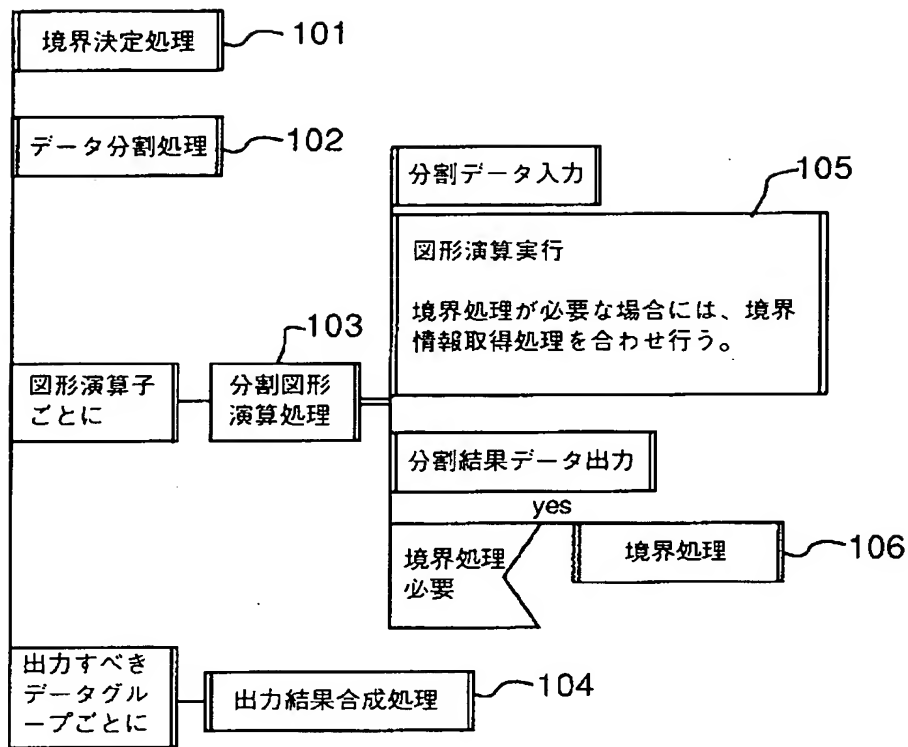


【図4】



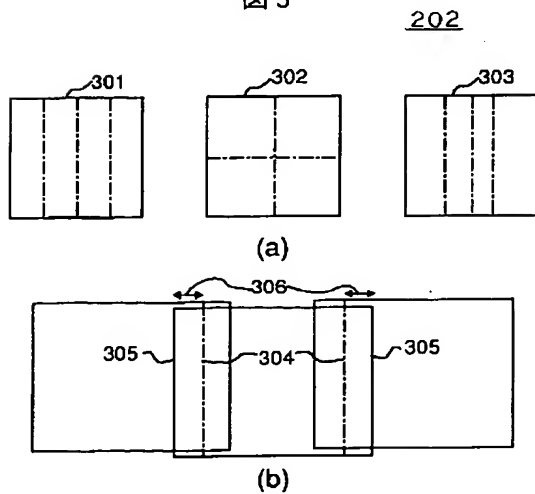
【図1】

図1



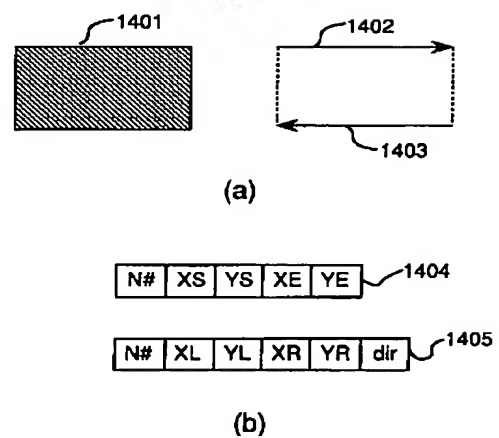
【図3】

図3

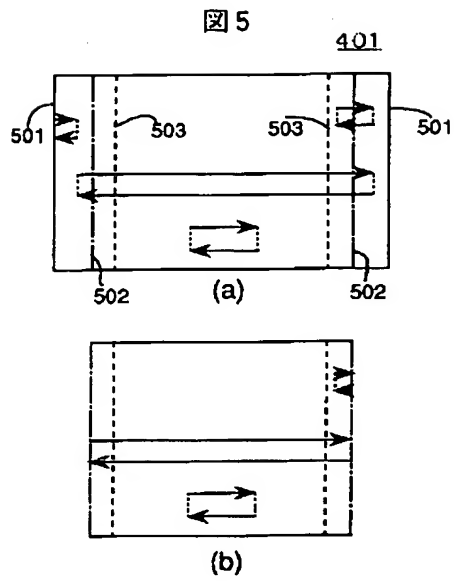


【図14】

図14

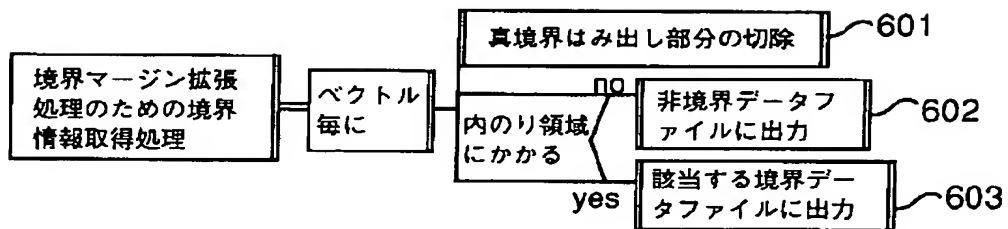


【図5】



【図6】

図6



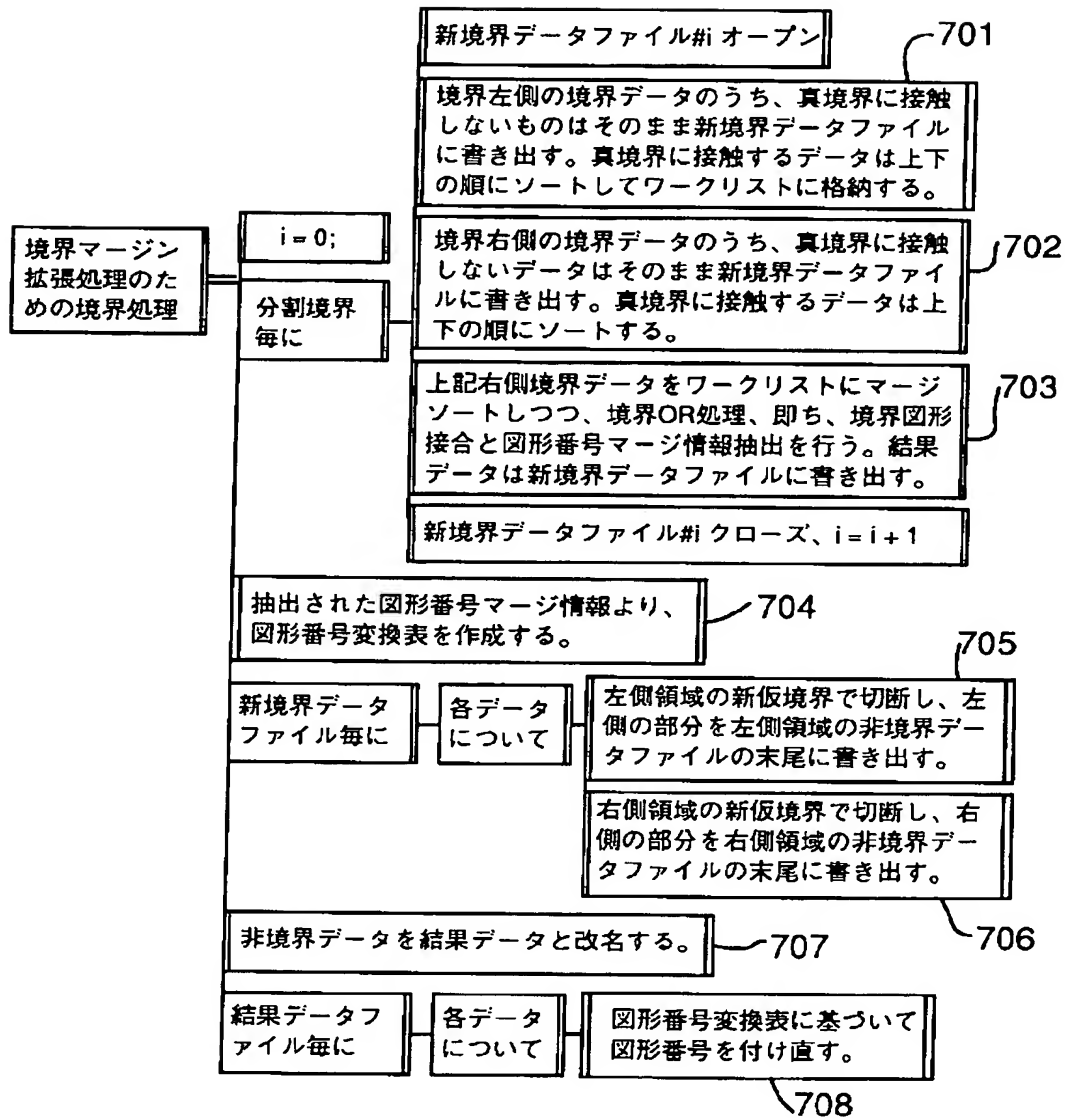
(a)



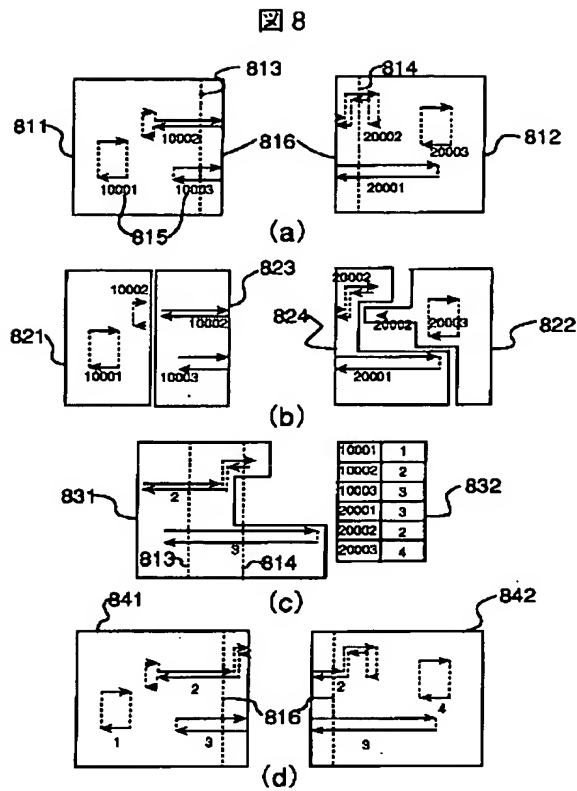
(b)

【図7】

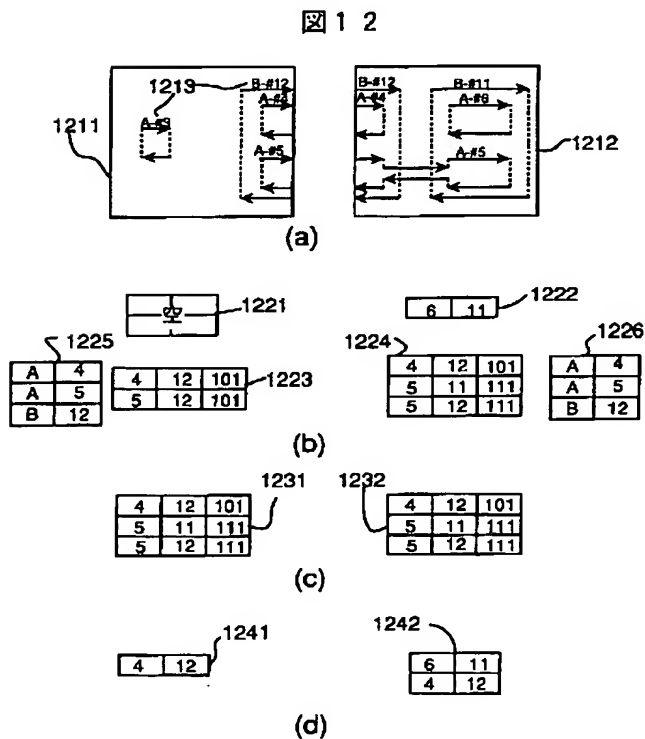
図 7



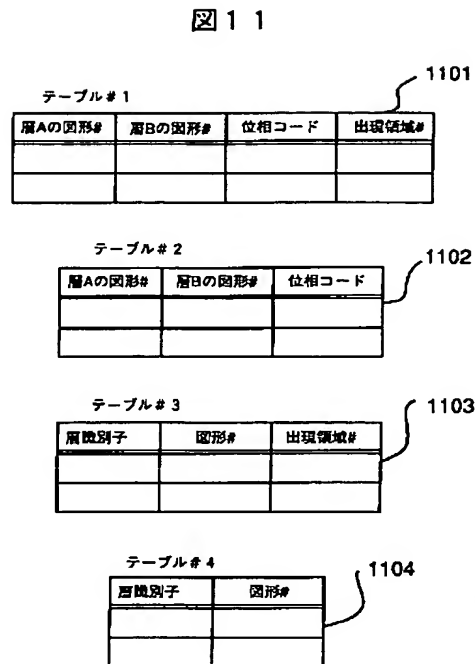
【図8】



【図12】

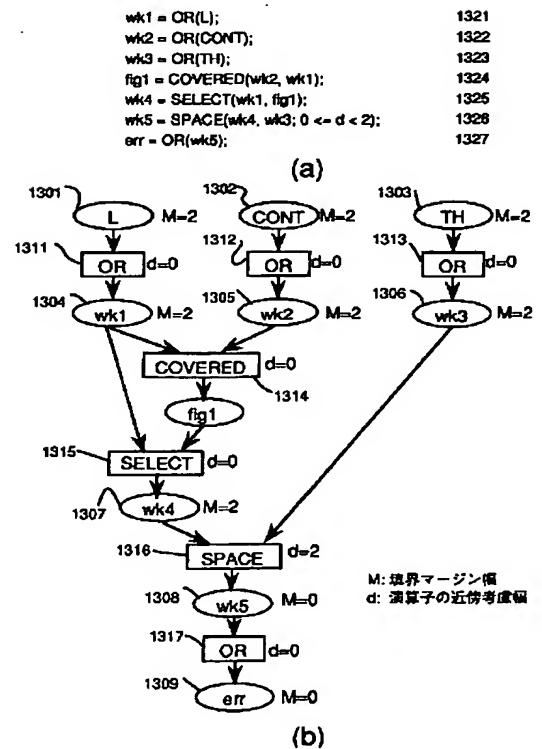


【図11】

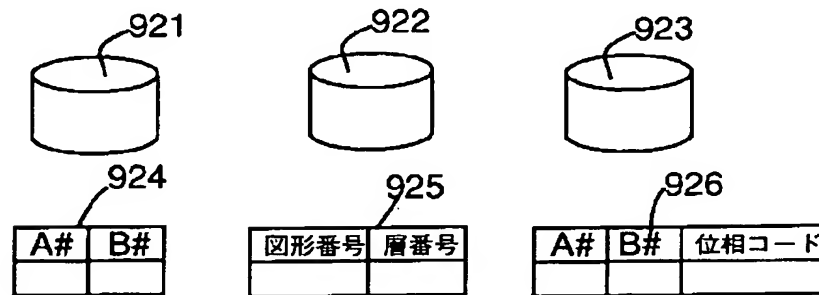
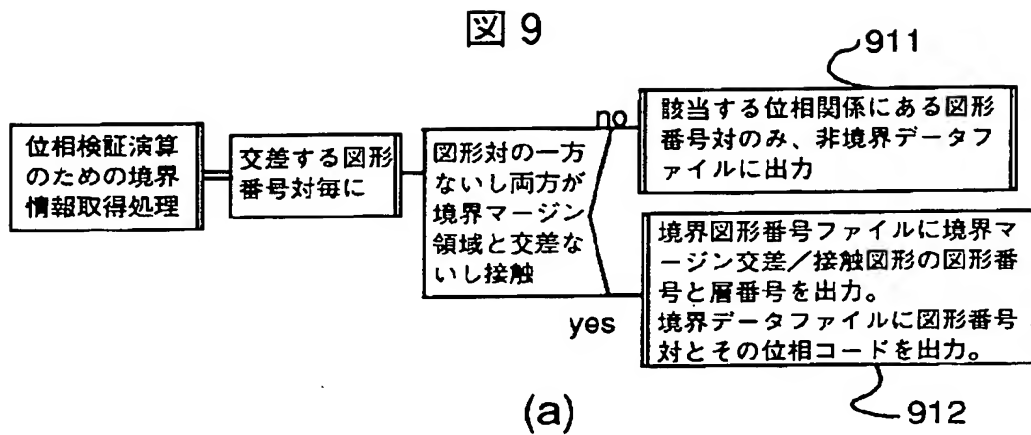


【図13】

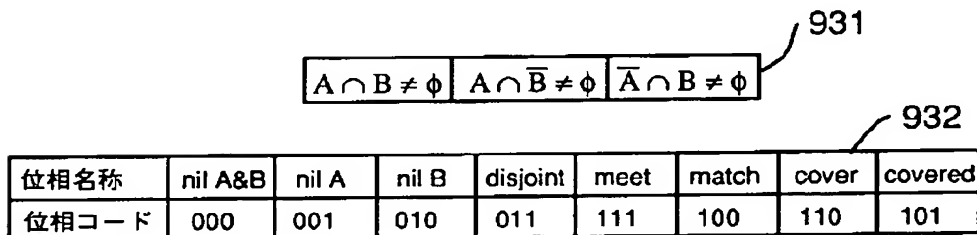
図13



【図9】



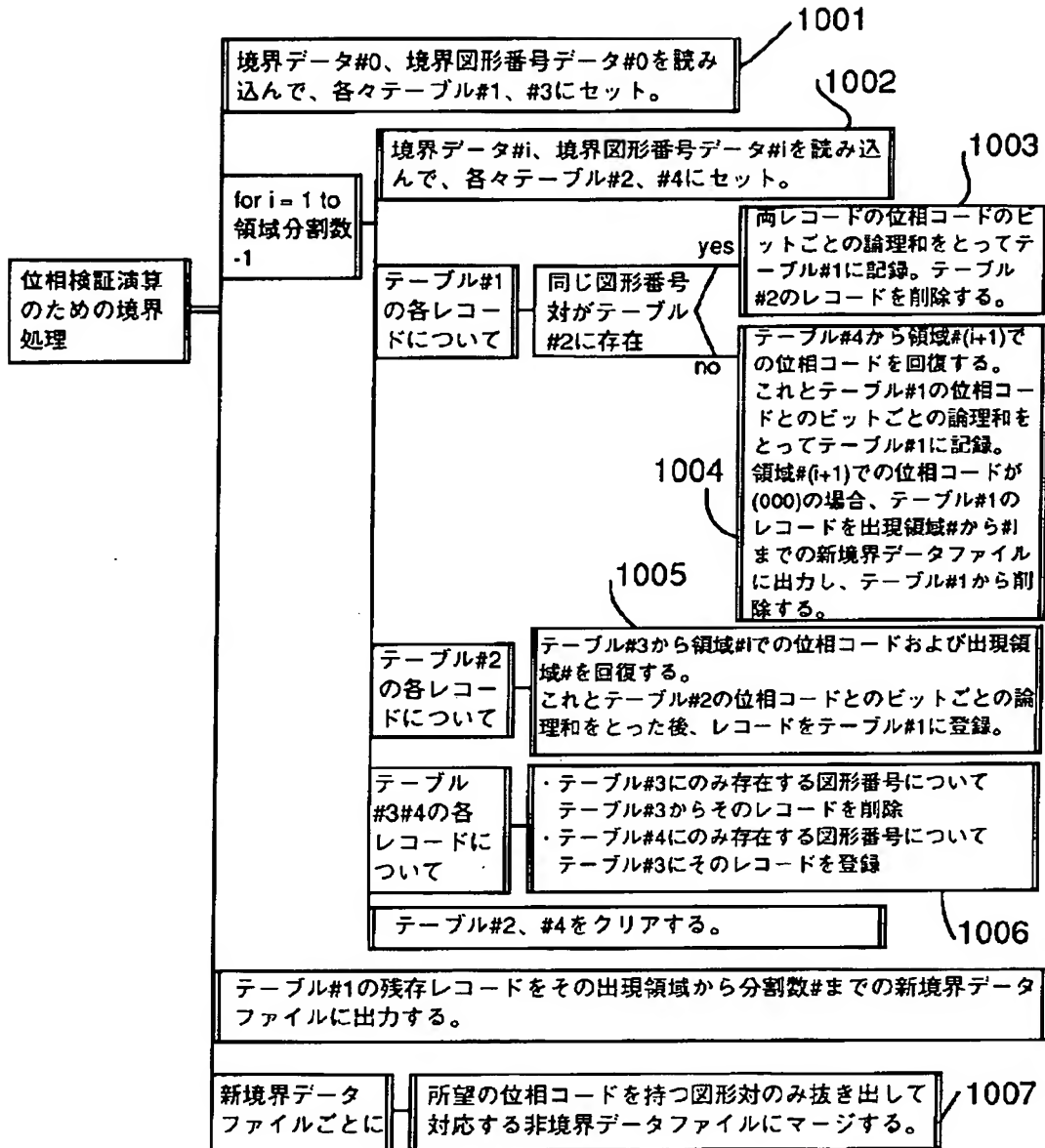
(b)



(c)

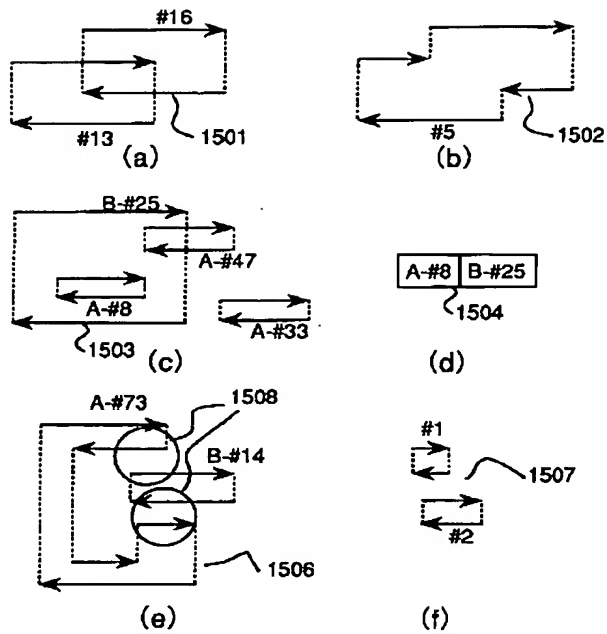
【図10】

図 10



【図15】

図15



【図16】

図16

